Cryptographic Primitives and Protocols for MANETs

Jonathan Katz
University of Maryland
Fundamental problem(s)

- How to achieve secure message authentication / transmission in MANETs, when:
  - Severe resource constraints make traditional cryptographic approaches infeasible/undesirable
  - Potential compromise of a significant number of nodes/keys must be taken into account
  - Adversary may have significant control over communication channels

- Design cryptographic primitives/protocols
  - Formal models
  - Rigorous proofs of security
Selected results

- “Secure Network Coding over the Integers,” PKC 2010
- “Authenticated Broadcast with a Partially Compromised PKI,” SSS 2010
- “Public-Key Cryptography Resilient to Continual Memory Leakage,” FOCS 2010
- “Provably Secure Location Verification,” in submission
Secure Network Coding over the Integers

R. Gennaro, J. Katz, H. Krawczyk, and T. Rabin
Network coding

- Routing mechanism whereby intermediate nodes modify packets in transit

- Advantages
  - Can increase data throughput
  - Better resilience to packet loss
  - No centralized control
Linear network coding

- View a file F as a sequence of vectors $v_1, \ldots, v_m$, where $v_i = (v_{i1}, \ldots, v_{in}) \in F^n$, typically $m << n$
- Define the “augmented” vectors $w_i = (0\ldots010\ldots0 v_i)$
- Transmit the $w_i$
- A node receiving vectors $y_1, \ldots, y_t$ chooses random weights $c_1, \ldots, c_t \in F$ and transmits $\sum c_i y_i$
  - Weights can be pre-assigned, or randomly chosen
- Given enough linearly independent vectors, receiver can recover the $v_i$
Example

recover a, b

recover a, b
File recovery: a note

- Recall that to recover the file, the receiver must obtain \( m \) linearly independent vectors.
- For random network coding, the exact probability with which the receiver recovers the file (the \emph{decoding probability}) \emph{depends} on two things:
  - Network size/topology
  - Field size \(|F|\)
- Experimentally, \(|F| \approx 2^8\) suffices.
Secure network coding

- Authentication is critical
  - A single incorrect vector can “pollute” many others
- How to authenticate the transmitted vectors?
  - Signing each packet does not work (since packets are modified *en route*)
  - Signing the entire file does not work either
Example

S

D1

recover a, b'

D2

recover a’, b

a

a’

a’+b

a’+b

b

b
Prior work

- Information-theoretic approaches
  - Impose a *bound* on the number of links/packets that the adversary can eavesdrop/modify
  - High communication overhead

- Cryptographic approaches...
  - Defining security
  - Homomorphic signatures [CJL06, BFKW09]
  - Homomorphic hashing [KFM04, ZKMH07, BFKW09]
Defining security

- Source generates (pk, sk); pk given to adversary
- Adversary can repeatedly request signatures on files
  - Adversary outputs (fid\textsubscript{i}, V\textsubscript{i}), where V\textsubscript{i} \subseteq F^n of arbitrary dimension m \leq n; the \{fid\textsubscript{i}\} are distinct
  - Adversary given signature \sigma_1, ..., \sigma_m on “file” V\textsubscript{i}
- Adversary outputs (fid\textsuperscript{*}, y, \sigma)
  - Adversary succeeds if Vrfy\textsubscript{pk}(y, \sigma) = 1 and either (1) fid\textsuperscript{*} \neq fid\textsubscript{i} for all i, or (2) fid\textsuperscript{*} = fid\textsubscript{i} but y \not\in V\textsubscript{i}
Homomorphic signatures

- Signature scheme with the following property:
  - Given signatures $\sigma_1, \ldots, \sigma_t$ on $y_1, \ldots, y_t$, anyone can compute a valid signature $\sigma$ on $y = \sum c_i y_i$ (for arbitrary, known weights $\{c_i\}$)
  - But impossible to forge a valid signature on any vector $y' \notin \text{span}(y_1, \ldots, y_t)$
- “Paradoxical” notion of security: “forgeries” allowed, but only specific kinds of forgeries…
- Can be used directly for secure network coding!
State-of-the-art [BFKW09]

- Constructions based only on *bilinear maps*
  - Computationally expensive (relatively speaking)
- Network coding done over *large* fields
  - Consequence of cryptographic parameters
  - Significant impact in bandwidth usage
    - E.g., when using 160-bit fields overhead there is a 20x blowup (due to network coding alone!)
This work

- First *RSA-based* homomorphic signatures
  - (Prior construction of Yu et al. [Infocom08] is shown to be insecure)
- Due to algebraic properties of RSA, network coding must be done over the *integers*
  - We analyze the decoding probability in this case
- As a consequence, better efficiency/bandwidth
  - Network coding over the integers can also be applied to schemes of [BFKW09]
- Also RSA-based homomorphic hashing
RSA-based homomorphic signatures

- Public key: Modulus $N$ (product of safe primes), public exponent $e$, generators $g_1, \ldots, g_n \in \mathbb{QR}_N$
- Signing:
  $$\text{Sign}(v_1, \ldots, v_n) = \left( \prod_j H(fid,j)^{u_j} \prod_i g_i^{v_i} \right)^d \mod N$$
- Verification: Obvious
- Homomorphic(?)
Is it homomorphic?

- \( \text{Sign}(v) \times \text{Sign}(w) \mod N = \text{Sign}(v + w \mod \phi'(N)), \)
  where \( \phi'(N) = |QR_N| = \phi(N)/4 \)

- **Problem**: intermediate nodes don’t know \( \phi'(N)! \)
  - If they do, scheme is broken

- **Solution**: work over the integers!
  - \( \text{Sign}(v) \times \text{Sign}(w) \mod N = \text{Sign}(v + w) \)
Difficulties

- Working over the integers introduces several complications that need to be addressed
  - Effect of decoding probability?
  - Coordinate growth
  - Additional attacks? ($\mathbf{v}$ and $\mathbf{v} + \phi'(N) \cdot \mathbf{w}$ have the same signature)
Protocol

- Each node acts as follows:
  - Given $y_1, \ldots, y_t$, choose random $c_1, \ldots, c_t \in Q$ and transmit $\Sigma c_i y_i$, where $Q$ is some specified set of integers

- Reconstruction:
  - Given vectors $(u_1 | v_1), \ldots, (u_n | v_n)$, form matrix $U$ and see if $U$ is invertible over the integers
Choosing Q

- We want Q to contain small integers
  - Coordinate length (and thus bandwidth overhead) depends on maximum value in Q
- Also want Q “large enough” to ensure correct decoding w.h.p.
- Fundamental lemma:
  - If $Q = \{0, \ldots, q-1\}$ for prime $q$, then decoding probability over the integers is at least as good as decoding probability over $\mathbb{F}_q$
- Suffices to take $q \approx 2^8$
Coordinate growth?

- Note that (because no modular reduction applied) size of coordinates grows at each hop
  - If the maximum in-degree is $m$, coordinates grow by at most $\log (mq)$ bits with each traversed node
- For networks with moderate path lengths and in-degree, bandwidth overhead is less than prior cryptographic solutions using 160-bit fields
Additional attacks?

- We assume the maximum path length in the network is known.
- In that case, there is a known upper bound on the magnitude of coordinates.
  - Signatures rejected if coordinates are too large.

- Theorem: This defines a secure network coding signature scheme based on the RSA assumption in the random oracle model.
Efficiency

- Shorter public key
- For “reasonable” network topologies…
  - Roughly 2-5x reduction in bandwidth overhead
  - 10-20x improvement in computational efficiency for homomorphic signature combination; similar improvement for signature verification
Ongoing work

- Better understanding of decoding probability
  - Can $q$ be reduced further?

- Simulations to estimate performance

- Better understanding of realistic network topologies
  - More accurate measurements of performance
Authenticated Broadcast with a Partially Compromised PKI

D. Gordon, J. Katz, R. Kumaresan, and A. Yerukhimovich
Authenticated broadcast

- Network of $n$ nodes, point-to-point communication between every pair of nodes
- **Broadcast**: one user (the *dealer*) wants to send a message $m$ to all other users
  - If the dealer is honest, all (honest) users receive $m$ regardless of adversarial behavior
  - If the dealer is dishonest, all (honest) users receive the same message
  - Fundamental problem in distributed computing
Classical results

- \( t \) = number of corrupted parties
- If \( t < n/3 \), broadcast is possible with no setup, no assumptions [PSL80]
  - This threshold is optimal for this setting
- For \( t < n \), broadcast is possible assuming a PKI and digital signatures [DS83]
  - “Authenticated broadcast”
What if keys are compromised?

- Classical solutions assume nodes are either *honest* or *corrupt*
  - Secret keys of honest parties assumed uncompromised
  - Nodes whose keys are compromised are treated as corrupted!
- We introduce a third category: honest users whose keys are compromised
  - Can we provide guarantees for such nodes also?
Our results

- We fully characterize when this problem is solvable
- Let $t_a$ = number of corrupted parties; $t_c$ = number of honest parties whose keys are compromised

**Feasibility**: if $2t_a + \min(t_a, t_c) < n$, authenticated broadcast is still possible
  - By an efficient protocol

**Impossibility**: authenticated broadcast is *impossible* if $2t_a + \min(t_a, t_c) \geq n$
Leakage-resilient cryptography
Traditional cryptography

- Traditional cryptography (whether symmetric-key, signatures, or public-key encryption) inherently relies on the existence of secret keys
  - Completely random
  - Completely unknown to the adversary

- What happens if these assumptions are violated?
(Partial) key compromise?

- An adversary may be able to get partial information about keys
  - Side-channel attacks
  - Non-uniform key generation
  - Same key used for several purposes
  - Partial compromise/virus
  - Cold boot attacks

- For traditional cryptosystems, *nothing* about security can be claimed in any of the above cases!
Leakage resilience

- Augment the model by allowing the adversary to obtain $f(sk)$, for any bounded $f$ of the adversary’s choice
  - E.g., adversary can get 90% of the secret key

- Prove security even if that occurs!
  - Leakage-resilient cryptosystem
Results

- First leakage-resilient signature schemes [Asiacrypt 2009]
  - Efficient schemes, secure for constant fraction of key compromise
  - Schemes from general assumptions, secure for leakage of almost the entire key
- Continual leakage resilience [FOCS 2010]
  - Secret key is updated over time; adversary can leak a *unbounded* amount of information overall
  - First constructions of public-key encryption and digital signature schemes
Thank you!